MICROSOFT.

WE SET THE STANDARD. WE ARE THE STANDARD.

MICROSOFT HAS CHALLENGING CAREERS FOR COMPUTER PROGRAMMERS.

We set the standard in microcomputer software.

If computer programming is your career choice, explore Microsoft and see what we have to offer. Microsoft has been around quite a while in this young industry...dating back to the advent of the microcomputer in 1974. Our first product, Microsoft® BASIC, is the most widely used software in the world: there are over one million installations already.

Microsoft not only has grown with the microcomputer industry, we've continued to stay ahead, setting the software standard. In every aspect of software production —research, design, development, documentation, technical support and marketing—we're at the forefront of the industry.

Our history is peppered with "firsts"—first independent supplier of microcomputer software; first software developer to offer high level languages (BASIC, COBOL, FORTRAN and Pascal) for microcomputers; and the first company to develop a family of user-friendly, system software products designed —from concept through production—to be integrated programs.

Microsoft is firmly committed to internal product development and technical excellence for our software. In 1974 the company was founded by systems programmers. Since its inception, Microsoft's business has been software. We're specialists in the field of superior microcomputer programs. Today nearly half of our company staff still is comprised of technical personnel developing new products and enhancing existing product lines.

As a strong, dynamic and rapidly growing software company, Microsoft is involved in software activities on many fronts:

▪ Development of traditional system software.

Microsoft provides operating systems, languages and utilities to OEM clients that fill a Who's Who in Computer Hardware Directory—including IBM, Apple Computer, Inc., Radio Shack, Texas Instruments, Intel and Tektronix. Our Consumer Products Division is a major force in marketing our software tools through retail outlets.

▪ Development of end-user system software.

There is a growing need for system software that laymen can use to define their own programs. Recognizing the product potential in this marketplace, Microsoft is committed to fulfilling this need with versatile, exciting "generic" software that runs on widely used personal computers like those marketed by Apple, IBM and Radio Shack.

▪ Research and development.

As a leader in the microcomputer software industry, Microsoft is actively involved in research. We're working on industry advancements in areas such as local networking, artificial intelligence and sophisticated graphics.

We have the right equipment and environment to conduct software R&D:

- Microsoft software is written on top mainframes: DEC® 2060 running TOPS-20, two PDP-11's and Vax 11/750 running XENIX™ OS (modified UNIX™ 3.0 operating system, licensed by Bell Laboratories). Then the software is downloaded to over 50 different microcomputer systems from American, European and Japanese manufacturers.
- Our software is tested, evaluated and refined on the latest microcomputer hardware, including 16- and 32-bit micros (Z8000, 8086 and 68000).
- We use excellent software tools such as high level languages (our programmers work and write in languages like C and Pascal) and operating systems we've developed.
- Corporate support, enthusiastic leaders and a commitment to improving the quality and capabilities of software also contribute to an excellent development atmosphere.

In each of these areas you'll have an opportunity to work closely and informally with some of the leading software authorities in the world—the people at Microsoft.

Opportunities at Microsoft—an outlet for your creativity and energy.

At Microsoft we make things happen. Here you'll be part of the development effort to bring state-of-the-art technology to the "real" world. We work on projects that come to fruition today. And our people are influencing the way software will look tomorrow.

Our OEM customer base includes leading manufacturers who value our input. As a Microsoft programmer, you'll have the chance to work with their latest hardware. Often, we help OEM's add hardware features to their machines which in turn enable us to create more powerful software for them. You'll also have an opportunity to give your technological input for these prototypes AND work on the software that will run on them.

Look to Microsoft for innovations like these:

■ Interactive System Software.

Multi-Tools™ is a series of end-user tools that bridges the steps between higher level system software and consumer-oriented applications software. With Multi-Tools, the end-user has straightforward, user-friendly, and intuitive software that allows him to create programming solutions specific to his projects without learning a computer language.

■ XENIX Operating System.

Another major Microsoft project is the XENIX operating system. An enhanced version of the licensed Bell Laboratories' Version 7 UNIX operating system for microcomputers, XENIX is Microsoft's answer to a high level OS for 16-bit micros. It has these powerful features:

- High portability so programs don't have to be rewritten.
- Flexibility to handle multiple processor installations.
- A versatile programming environment that includes high level languages, an operating system and program development tools. XENIX simultaneously supports tasks at all these levels so users can dynamically connect and reconnect utility processes in any way, giving the programmer great flexibility.

Functionality and support are key words with XENIX OS. "Exciting" is the best way to describe its potential and the work that will be done with it.

■ Advanced Compiler Technology.

We're developing state-of-the-art compiler technology at Microsoft. As new microprocessors are introduced, we are the ones who work on new compilers to run on them. Microsoft compilers used advanced optimization techniques such as common sub-expression elimination, peephole optimization and common optimizers for different language front ends.

We're working with compilers we've developed for the following processors:

- Pascal compiler written in Pascal with many extensions for 8080, 8086, Z8000 and 68000.
- COBOL-74 compilers for 8080, 8086.
- C compilers written in C for 8086, Z8000 and 68000.
- FORTRAN compilers for 8080, 8086, Z8000 and 68000.
- BASIC compiler for 8080 and 8086.

Microsoft maintains its dominant role in the software industry by staying on the cutting edge of product advancement—anticipating the software needs of the future and the development challenges that software companies must meet. Opportunities are broad and far-reaching for our programmers.

Software portability is one of the significant challenges facing the Microsoft programmer. As the demand for software continues to escalate, it becomes essential that any software package be available for many different computers. Instead of writing programs for one specific machine, Microsoft programmers are creating software that runs on a dozen or more computers.

Ours is a "software is our business" environment. Microsoft was founded and is run by system software developers who understand your needs. We offer:

- Participation on a small development team.
- Projects on a one task/one person basis.
- The chance to learn from leading software professionals.
- Personal satisfaction and professional recognition. Microsoft products go places. They're used by thousands of people around the world. Your work here can have high visibility and real impact on the microcomputer software industry.
- The opportunity to be part of a small, growing company that is the force in microcomputer software.

HERE IS A TYPICAL PROGRAMMING ASSIGNMENT YOU COULD ENCOUNTER AT MICROSOFT.

How would you handle it?

Writing transportable programming is a very important aspect of the software business. The more portable the software, the larger the potential market for it.

The requirements of transportability create interesting situations where theoretical and practical problems appear together.

A portable program by definition is one that does not depend on a particular machine architecture. This raises the following paradox:

If the code does not depend on an architecture, how can it exploit special capabilities efficiently? Clearly, a special capability must be treated with special code that is not transportable. The challenge then is to keep as much of the program logic in the portable—or "abstract"—portion as possible but still provide an information-rich environment for the machine-specific portion so that it can work efficiently.

An example for this is a debugger for an intermediate language interpreter, such as one that might be used for Pascal. An optimal debugger should have the following properties:

1. No interference with the environment of the program being debugged (subject program).
2. Ability to inspect and alter the complete state of the subject.
3. The ability to interact with the subject program at a high repetition rate; for example: for tracing, conditional breakpoints or performance measurements.

The implementation of the optimal debugger requires independent processes that can co-reside in the central memory using some sort of hardware address mapping. None of today's popular micros can do this. The engineering solution to the original problem retreats from the optimum to accommodate the reality of various hardware architectures.

A system which dumps the complete machine state on backing storage at every call to the debugger will generally satisfy (1) but go against (3).

On the other hand, a debugger sharing address space with the subject program will seriously interfere with the available space while satisfying condition (3).

The relative merits of the different trade-offs depend not only on the available hardware (speed and size of backing store, size of address space) but also on the phase of program development: initially there will be many bugs to be found, even with simpler test input. In later stages of development performance measurements will be important. During program maintenance, the ability to recreate problems that may occur under fully stressed operation conditions becomes most important.

In this example, the key to portability is the design of the interface used by the debugger to inspect and alter the subject program and the method of transfer of control.

With the proper definition the "non-portable" portion will be relatively small compared with the debugger proper that includes the user interface and language specific knowledge. If the interface is able to express some specific information about the semantics of a control transfer (e.g. continue while location 1202 does not equal 0) new compromise solutions in addition to the two extremes described above, also may be obtained.

In conclusion, portability considerations force us to partition the problem into logical modules that "hide" machine or implementation—specific details. Different configurations of the modules can be useful even within a single system to overcome different limitations for different applications.

What are your ideas concerning debuggers? How would you define the interface between the debugger and the subject program? Write to us with your thoughts about this problem and we will share our ideas with you.

Send your comments to Jo Ann Rahal, Microsoft Corporation, 10700 Northup Way, Bellevue, WA 98004.

WE WANT THE BEST PEOPLE AT MICROSOFT.

Integral to our success are our people. Key roles are played by our programmers. We don't want good programmers...we need outstanding programmers.

We seek individuals with these qualifications:

- Potential for professional growth and a demonstrated interest in software through relevant experience:
  - pertinent summer employment
  - work at a campus computer center
  - participation in a graduate research program
  - other employment as a working professional
- A well-established base of technical knowledge plus an eagerness to learn and grow.
- Personal attributes such as self-drive, initiative, enthusiasm and the ability to work independently as well as part of a team.

A CLOSER LOOK AT MICROSOFT

Microsoft has come a long way since it was founded in 1974 as a two-person operation with one product. Since then it has become a growth-oriented company with a rapidly expanding product line that is distributed through OEM channels and retail dealers.

The product line ranges from operating systems and system software to end-user tools and applications programs.

There is a coherency of direction at Microsoft from development of new products to complete company support of these projects.

We decide which products we want to pursue technically; we design and implement them in a professional manner; and then we market our software to microcomputer users around the world.

Key events in Microsoft's history.

February, 1975—
BASIC completed and sold to Microsoft's first customer, MITS, manufacturer of the ALTAIR personal computer.

July, 1976—
BASIC refined, enhanced and sold to other customers including General Electric, NCR and Citibank.

July, 1977—
Microsoft began selling its second language product, FORTRAN.

December, 1977—
1977 year end sales: $1 million. Employees: 5.

June, 1978—
Microsoft began selling its third language product, COBOL.

Fall, 1978—
BASIC licensed to Radio Shack and Apple. Microsoft initiated sales to Japan.

December, 1978—
1978 year end sales: $2 million. Employees: 13.

Spring, 1979—
Microsoft introduced BASIC-86.

June, 1979—
Microsoft founded Consumer Products Division dedicated to retail and end-user customers.

December, 1979—
1979 year end sales: $4 million. Employees: 25.

January, 1980—
Consumer Products Division shipped its first product, Level III BASIC.

February, 1980—
Microsoft licensed the UNIX operating system (Bell Labs) and the XENIX group was formed.

May, 1980—
Microsoft introduced BASIC Z8000.

August, 1980—
Microsoft introduced SoftCard,™ a peripheral for the Apple® II. Sales of 25,000 recorded during its first year of availability.

December, 1980—
1980 year end sales: $8 million. Employees: 40.

February, 1981—
Microsoft formed the End-User System Software group.

August, 1981—
Introduction of the IBM® Personal Computer which uses Microsoft's MS™-DOS operating system, BASIC, Pascal and other Microsoft products.

November, 1981—
Microsoft announced The Manager Series™ and Multiplan,™ first product in the Multi-Tools series.

December, 1981—
1981 year end sales: $16 million. Employees: 125.

Benefits we offer:

- Flexible working hours.
- Stimulating work environment where your value as an employee is recognized with competitive compensation, opportunities for advancement and challenging projects.
- Health care and life insurance plans.

Multiplan, Multi-Tools, SoftCard, MS and XENIX are trademarks and Microsoft is a registered trademark of Microsoft Corporation.

DEC is a registered trademark of Digital Equipment Corporation

The Manager Series is a trademark of The Image Producers, Inc

IBM is a registered trademark of
International Business Machines Corporation

Apple is a registered trademark of Apple Computer, Inc

UNIX is a trademark of Bell Laboratories

©Microsoft Corporation, 1982

Pacific Northwest lifestyle:

Name your sports, recreation, cultural or social interest and it's available. Choose your favorite vacation setting and it can be found here.

Microsoft is located in Bellevue, Washington, just 10 miles and one floating bridge away from downtown Seattle, hub of the region's major sports, cultural and commercial activities.

Outdoor environments ranging from mountains, ocean beaches, desert, rainforest, fresh water lakes and rivers are within easy reach.

Known for its friendliness, Seattle is an easy-going, heterogeneous mix of people from all parts of the United States and numerous foreign countries.

It's a great place to live. You'll fit in.

AT MICROSOFT THERE'S ONLY ONE STANDARD: THE BEST. JOIN US.

MICROSOFT®

Microsoft Corporation
10700 Northup Way
Bellevue, WA 98004
206-828-8080